Amendments to the Specification:

Please replace paragraphs [0001], [0035], [0036], [0038] and [0047] with the following amended paragraph:

[0001] This application is related to (1) U.S. Patent Application No. 10/104,267 filed March 22, 2002, and entitled "Adaptive Connection Routing Over Multiple Communication Channels," (2) U.S. Patent Application No. 10/104,245 filed March 22, 2002, and entitled "Abstract User Interface Manager with Prioritization," and (3) U.S. Patent Application No. 10/805,065 ~~(Attorney Docket No. SUNMP177)~~ filed on the same day as the instant application, and entitled "Method and Apparatus for User Interface Manager," each of which are incorporated herein by reference in their entirety for all purposes.

[0035] Still referring to Figure 4, operating system layer 122 sits above hardware layer 120. Java virtual machine (JVM) layer 124 sits on top of operating system (OS) layer 122 and open services gateway initiative (OSGI) layer 126 sits on top of the JVM layer. It should be appreciated that the standard for JVM layer 124 includes Java 2 Platform Micro Edition (J2ME), Connected Device Configuration (CDC), Foundation Profile, Personal Profile or Personal Basis Profile. One skilled in the art will appreciate that J2ME Foundation Profile is a set of APIs meant for applications running on small devices that have some type of network connection, while J2ME CDC Personal Profile or Personal Basis Profile provides the J2ME CDC environment for those devices with a need for a high degree of Internet connectivity and web fidelity. The standards for each of the layers of the stack are provided on the right side of client side implementation 121. In particular, OSGI 126a, J2ME CDC 124a, OS 122a, and embedded board 120a are standards and to the left of the standards are

examples of actual products that implement the standards. For example, OSGI 126a standard is implemented by Sun's Java Embedded Server (JES) 2.1 126b, J2ME 124a standard is implemented by Insignia's Virtual Machine 124b, OS 122a is implemented by Wind River's VxWorks real time operating system 122b, and embedded board 120a is an embedded personal computer based board such as Hitachi's SH4 120b. It should be appreciated that the actual products are exemplary only and not meant to be limiting as any suitable product implementing the standards can be utilized.

[0036] Carlets 132 of Figure 4, have access to each layer above and including OS layer 122. Application program interface (API) layer 130 is the layer that carlets use to communicate with the JPS ~~JTC~~. Service provider interface (SPI) layer 128 is a private interface that managers have among each other. One skilled in the art will appreciate that OSGI layer 126 provides a framework upon which applications can run. Additional functionality over and above the JVM, such as lifecycle management is provided by OSGI layer 126. It should be appreciated that the open services gateway initiative is a cross industry working group defining a set of open APIs for a service gateway for a telematics systems. These APIs consist of a set of core framework APIs. In order to deploy services and their implementations OSGI defines a packaging unit called a service bundle. A service bundle is a Java Archive (JAR) file containing a set of service definitions along with their corresponding implementation. Both infrastructure services and carlets are deployed as service bundles. Some of the functionality for arbitrating, controlling and managing devices and resources, e.g., speakers, cell phones, etc., by OSGI layer 126. However, one skilled in the art will appreciate that a separate arbitration service may also be required. As used

herein, a carlet is a Java application. For each function or task to be processed on the client side or between the client and server sides, a carlet is invoked to manage the operation. In this manner, carlets can be independently written, tested, and launched for use on a telematics system. By way of example, a carlet can be written to control or monitor the activity of automobile components (e.g., tires, engine oil, wiper activity, steering tightness, maintenance recommendations, air bag control, transmission control, etc.), and to control or monitor applications to be processed by the telematics control unit (TCU) and interacted with using the on-board automobile monitor. As such, specialized carlets can be written to control the audio system, entertainment modules (e.g., such as on-line games or movies), voice recognition, telecommunications, email communications (text and voice driven), etc. Accordingly, the type of carlets that can be written is unlimited. Carlets may be pre-installed or downloaded from a sever. A carlet may or may not have an API which may be invoked by other carlets and it may or it may not have running threads of its own.

[0038] Still referring to Figure 5, when a particular carlet application 402 in a plurality of carlet applications 402a-402n is requested, the carlet will communicate with the stream manager 419 and request that a connection be established. In response, the stream manager 419 will request a connection object (Conn. OBJ) 418a from the data multiplexer and flow controller 415. Once a channel satisfying the request is available, the data multiplexer and flow controller 415 will return a connection object (Conn. OBJ) 418b back to the carlet. Thus, a communication link 432 is established between the carlet application 402 via the connection objects 418a and 418b of the data multiplexer and flow controller 418 ~~415~~. In one embodiment, the connection object 418a of the data multiplexer and flow

controller 418 has the ability to switch between channels 425 that are available to the communications framework 416 of the client side. Here, code contained in the policy manager enables selection of different channels depending upon availability, the type of communication desired, bandwidth requirements for a given data transfer or transfers, payment of a bandwidth fee, subscription level, etc.

[0047] Figure 11 is a simplified schematic diagram of status bar which further exemplifies the advantages of having a draw manager as a middle man in one embodiment of the invention. Here, the status bar is constantly being updated as some process is progressing. For example, the drawer manager is updating a display every 5 millisecond ~~%~~ but an application may be performing an update every 1 millisecond ~~%~~. Thus, the drawer manager is configured to optimize the display based upon the hardware and operating system for the computing environment in which the draw manager resides.